

~15~

PAT-NO: JP401057304A

DOCUMENT-IDENTIFIER: JP 01057304 A

TITLE: METHOD FOR ANALYZING CONTROL SYSTEM

PUBN-DATE: March 3, 1989

INVENTOR-INFORMATION:

NAME

MARUYAMA, HARUMI

TAKANO, TAIKO

KATAYAMA, YASUNORI

KATAOKA, HIDEO

INT-CL (IPC): G05B017/02

US-CL-CURRENT: 700/182

ABSTRACT:

PURPOSE: To express a controlled system to be complexly operated by arranging a control element on the screen of a display device and registering optional control operation as a new control element.

CONSTITUTION: A control system CAD system is constituted of a step for initializing various variables required at the time of starting a CAD, a step for executing the input of a model for forming a block diagram indicating the flow states of signals in the control system based on various input processing, a step for executing the compile of each program defined by a user as the processing contents of a program block and executing link processing with programs in the whole control CAD system, a step for executing simulation, and a step for setting up various display specifications and displaying a graph. Optional control operation is described and registered as a new control element. Consequently, the restriction of a function such as a linear element provided by the system is removed, so that the analyzable range of the control system can be expanded.

COPYRIGHT: (C)1989,JPO&Japio

6/26/06, EAST Version: 2.0.3.0

⑫ Int. Cl.

識別記号

庁内整理番号

⑬ 公開 昭和64年(1989)3月3日

G 05 B 17/02

7740-5H

審査請求 未請求 発明の数 1 (全14頁)

⑭ 発明の名称 制御系解析方法

⑮ 特 願 昭62-212798

⑯ 出 願 昭62(1987)8月28日

⑰ 発 明 者 丸 山 晴 美 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内
 ⑰ 発 明 者 高 野 た い 子 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内
 ⑰ 発 明 者 片 山 恭 紀 茨城県日立市久慈町4026番地 株式会社日立製作所日立研究所内
 ⑰ 発 明 者 片 岡 秀 雄 神奈川県横浜市戸塚区戸塚町5030番地 株式会社日立製作所ソフトウェア工場内
 ⑱ 出 願 人 株式会社日立製作所 東京都千代田区神田駿河台4丁目6番地
 ⑲ 代 理 人 弁理士 小川 勝男 外2名

明 細 書

1. 発明の名称

制御系解析方法

2. 特許請求の範囲

1. 予め設定された制御系の特定の動作を示す制御要素を画面上に対話的に配置し、前記画面上に配置された制御要素間を結線して制御対象のモデルを作成し、当該モデルを用いてシミュレーションを行う制御系解析方法において、任意の制御動作を記述し、前記記述された任意の制御動作を新たな制御要素として登録し、該登録された制御要素と前記予め設定された制御要素間で結線することを特徴とする制御系解析方法。
2. 特許請求の範囲第1項において、前記任意の制御動作はプログラミング言語により記述されることを特徴とする制御系解析方法。
3. 特許請求の範囲第1項において、前記登録された制御要素は、パラメータを持ち、計算部に該パラメータ値を参照又は書換えてできる制御要素とすることを特徴とする制御系解析方法。

3. 発明の詳細な説明

〔産業上の利用分野〕

本発明は制御系を解析する方法に係り、特に複雑な動作を行うシステムの解析に好適な制御系を解析する方法に関する。

〔従来の技術〕

従来の制御系解析方法は、「HITAC マニュアル プログラムダクト GRADAS 自動制御設計システム HICAD/AC 8090-7-049-10」に記載されている。

この制御系解析方法は、ブロック図をワークステーション上で作成し解析を行うものであり、線形要素又は一部非線形要素等のシステムが提供した機能の組合せてブロック図を作成する。

一方、制御対象は多入力、多出力で非線形性を有する場合があり、この場合前記システムが提供する機能の組合せでは表現できない。

〔発明が解決しようとする課題〕

上記従来技術は、汎用的な非線形性を有する制御対象を記述する点に配慮がされておらず、複雑

な動作を表現することができない問題があった。

本発明の目的は、任意の制御動作を新たな制御要素とすることにより制御対象の複雑な動作の表現を可能とし、制御系の解析可能な範囲を拡大することにある。

〔問題点を解決するための手段〕

上記目的は、予め設定された制御系の特定の動作を示す制御要素を表示装置の画面上に対話的に配置し、前記画面上に配置された制御要素間の結線を行い制御対象のモデルを作成し、シミュレーションを行う制御系解析方法において、任意の制御動作を記述し、前記記述された任意の制御動作を新たな制御要素として登録し、前記登録された制御要素と前記予め設定された制御要素間で結線を行うことにより達成される。

尚、前記任意の制御動作とは、予め設定された制御系の特定の動作以外の制御動作をいい、具体的には前記任意の制御動作はプログラム言語、電気回路図又はP&ID図により記述される。

〔作用〕

種表示仕様を設定しグラフを表示する結果の表示ステップ50から構成される。

前記ステップ10、50の実現方法は、例えば「HITACマニュアル プログラムダクト対話型図形制御プログラムIGCF CAD TCS /Hインターフェース編 8090-7-024-20(以下、IGCFマニュアルと呼ぶ)」の第10頁から第24頁に記載されている。

また、前記ステップ30の実現方法は、例えば「HITACマニュアル プログラムダクト VOS2 VOS3 最適化FORTRAN77,H&P FORTRAN77 使用手引8080-3-258-50」の第23頁から第41頁に記載されている。

前記モデル入力ステップ20の処理の流れを図3図に示す。

はじめにモデル入力命令が入力されると、命令の種類及び命令に必要な引数等を生成する前処理ステップ210を実行する。次に、前処理ステップ210から得られた命令が何であるかを判断するステップ220を実行し、以下各命令に応じて

任意の制御動作を記述し、新たな制御要素として登録する。それによって、線形要素又は一部非線形要素というシステムが提供した機械的制約がなくなるため、制御対象の複雑な動作の表現が可能となり、制御系の解析可能な範囲を拡大することができる。

〔実施例〕

以下、本発明の一実施例を図2図により説明する。ここでユーザが記述した任意の制御動作を登録する制御要素をプログラムブロックと呼ぶ。

制御系CADシステムは、CADを起動した時に必要な各種変数の初期設定を行うステップ10、次に各種入力処理によって制御系中信号の流れの様子を表わすブロック結線を作成するモデルの入力を実行するステップ20、ユーザがプログラムブロックの処理内容として定義した各プログラムのコンパイルを行い制御系CADシステム全体体のプログラムとのリンク処理を実行するステップ30、各種計算のための計算条件を定め計算を行うシミュレーションを実行するステップ40、各

処理を行う。

命令がブロック結線の構成要素であるシンボルを配置する命令である場合は、次のような処理が行われる。シンボル数が増加するため、新しいシンボルに付随する情報を格納する新しいメモリを、前記メモリの参照を容易に行わしむるために作成するシンボルマップに追加するステップ230が実行され、次にその新しいメモリに前処理ステップ210から必要な引数として得られた各情報を代入するステップ240が実行される。

命令がシンボル結線である場合は、次のような処理が行われる。前処理ステップ210から必要な引数として、入力端子側と出力端子側のシンボルの情報が得られている。結線によって、入力端子側のシンボルから出力端子側のシンボルへ信号が流れることが示されるため、それに合致するように入端子側と出力端子側のシンボルに付随する該当情報をそれぞれ更新するステップ250、260が実行される。

命令がプログラム編集の場合は、別ウィンドウ

で、ユーザが前記プログラムブロックの処理内容として定義したプログラムを組築するステップ270を実行する。

次にモデル入力の結果を出力する後処理ステップ290を実行し、次の命令が入力されると、前処理ステップ210から再びくりかえし実行する。

モデル入力終了命令が入力されると、判断ステップ220によつて、次に終了処理ステップ280を実行する。

前記ステップ210、290の実現方法は、例えば「IGPマニュアル」の第10頁から第24頁に記載されている。

次にモデル入力ステップ20において、前処理ステップ210から得られた命令がシンボル組築である場合に使用されるシンボルマップに含まれるメモリのデータ構造の概要を第4図(a)に、具体的な実現方法の一例であるリスト構造を第4図(b)に示す。

但し以後は簡略化のため、ポイントを省略した第4図(a)の如く、表現することとする。

引数変数には、種別変数の内容に応じた数値の値が格納される。本図は、FUNCが3つの引数P1、P2、P3をもつ例を示すものである。

伝達元変数には、BOXという名前のシンボルに情報を伝達するシンボルの名前FROM1、FROM2が格納される。

伝達先変数には、前記シンボルBOXが情報を伝達する先のシンボルの名前TOが格納される。

このような場合、FROM1、FROM2、TOというシンボル名に対応する3個のメモリがシンボル名BOXのメモリが含まれているシンボルマップの中に、同時に存在していることになる。

次に、種別がプログラムブロックである場合のシンボルマップに含まれるメモリのデータ構造の概要を第4図(c)に、具体的な実現方法の一例であるリスト構造を第4図(d)に示す。

但し、以後は第4図(e)と同様に第4図(f)の如く表現することとする。

種別が前記一般の場合には、ブロック図で用いられる基本記号や伝達関数や入力等なので、そ

れ1つのブロック図には、各1個の前記シンボルマップがわりあてられる。シンボルマップは、シンボルに付随する情報を格納するメモリから構成される。

シンボルマップに含まれるメモリの個数は、そのとき存在しているシンボルの数に等しく、シンボル組築(追加または削除)によつて、メモリの個数が増減する。

前記メモリは、名前変数、種別変数が各1個、種別変数の内容に応じて個数が決まる引数変数、および、はじめは零個であるが、シンボル組築によつて個数が増減する伝達元変数と伝達先変数から構成される。以下、第4図(a)を例に説明する。

名前変数には、シンボルの名称を表す唯一の文字列が格納される。本図においてはBOXが格納されている。種別変数には、一般の場合にはブロック図で用いられる基本記号、伝達関数、入力等の種類が格納される。ここでは、FUNCとする。

れに応じて引数変数、伝達元変数と伝達先変数の個数の範囲が決まっていた。

種別がプログラムブロックである場合には、ユーザがプログラムブロックの処理内容として定義したプログラムの属性をシンボルのメモリ中に保持する必要がある。そのため種別変数には、プログラム名のみならず、入力個数、出力個数、引数個数のメモリのアドレスがわりあてられ、各種が代入される。

例えば、第4図(a)では「プログラムブロックの処理内容はPROCという名前のプログラムであり、そのプログラムは2個の入力と1個の出力と1個の引数をもっている」とユーザが定義している場合であり、入力個数2、出力個数1、引数個数1が代入されているため引数変数、伝達元変数、伝達先変数の個数が、それぞれ1個、2個、1個となっている。

シミュレーションを行うステップ40を第6図に示す。

はじめに、シンボルマップから、計算の順序を

どを格納しておく各シンボルの計算順序及び前回値と今回値の配列を示す図（以下、計算配列を称す）を作成するステップ410を実行する。次に、ステップ420でユーザが入力したシミュレーション条件、計算を行う始めの時刻と終わりの時刻などから、計算時刻変数TIMEを初期化したり、計算回数Nや計算きざみDLTを設定する。次にステップ430で、初期計算時刻での値を求め、計算時刻、計算結果などを保存する結果表に格納する。

計算時刻変数TIMEをステップ450で計算きざみずつ増やししながら、その時刻TIMEにおける計算を行なうステップ460を実行し、計算結果を結果表に格納するステップ470を計算回数Nまで繰り返す。計算回数の判定はステップ440で行っている。

計算配列の作成を行うステップ410を第8図に示す。

計算配列は、個々のシンボルの計算順序などを格納しておく表である。

ステップ4140では、計算配列に未だ登録されていないメモリを選び、選んだ未登録のメモリの全ての仮置元のシンボルの値が決定済否かをステップ4150で判断する。ここで値が決定済であるシンボルとは、計算配列に登録済であるシンボル又は、初期値が自力で決まるシンボルである。

ステップ4150の条件を満たしている場合は、登録できる状態であるため、ステップ4160で、選んだ未登録メモリの名前を、計算配列に登録する。

ある計算時刻TIMEにおける計算ステップ4160を第7図に示す。

計算配列についてステップ4610で第1行から最終行まで処理したか否かを判定しつつ、最終行を処理するまで順次1行ずつ取り出すステップ4620を実行し、ステップ4630からステップ4690までで、その行のメモリの計算結果を求める処理を繰り返し実行し、最終行を処理した段階でステップ460が終了する。

はじめに、ステップ4110で、シンボルマップを順次調べて値が自力で決まるシンボルのメモリに含まれる名前を登録する。

値が自力で決まるシンボルとは、例えば種別が入力のシンボルである。

次にステップ4120で、シンボルマップ中のメモリが計算配列に登録済であるかを順次調べていき、全てのメモリが登録済になるまで、ステップ4130からステップ4160で未登録のメモリを登録できる状態になったかを順次調べ、登録できる状態であれば計算配列に登録する処理を繰り返し実行し、全てのメモリが登録済になった段階でステップ410の実行が終了する。

尚、ステップ4130から4160までの各ステップの関係を以下に説明する。

まず、ステップ4130でシンボルマップ中のメモリを順次調べていき、全てのメモリを試し終るまで、ステップ4140からステップ4160を繰り返し実行する。全てのメモリを試し終ると、ステップ4130へ戻る。

はじめに、ステップ4630で、計算配列から選ばれたある名前をもつメモリをシンボルマップから抽出し、全ての仮置元の名前を求め、計算配列から全ての仮置元の値を抽出する。

次に、ステップ4640で、シンボルマップから自分自身の引数値を抽出し、計算配列から自分自身の前回値を抽出する。

次に、シンボルマップ表から、そのシンボルの種別が何であるかを判断するステップ4650を実行し以下、各命令に応じて処理を行う。

種別がプログラムである場合は、次のような処理が行われる。

ステップ4660で、シンボルマップからそのシンボルの種別にわりあてられたプログラム名が何かを抽出する。次に、ステップ4670でその名前のプログラムを実行し計算結果を今回値に格納する。

種別が自由出力である場合は、ステップ4680で、ステップ4630で得られた仮置元の今回値を計算結果として今回値に格納する。

種別が突合せである場合は、ステップ4690を実行する。ステップ4630で得られた全ての伝搬元の今回値にステップ4640で得られた引数値、すなわちそれぞれの伝搬元に対応する位置の突合せ符号をつけて、加算した値を計算結果として今回値に格納する。

種別が上記以外の場合は、その種別に対応した処理を行い、選ばれたその行のメモリの計算結果を求める。

ステップ4610で計算配列について最終行まで処理したことが判定されると、ステップ460が終了する。

以上のような構成により第1図(a)に示すブロック線図を作成し、シミュレーションを行う場合を考える。

第1図(a)において、4は箱Fというシンボル名をもつ前記プログラムブロックのシンボルである。

第1図(b)は箱Fを拡大したイメージである。箱Fというシンボル名のプログラムブロックの処

0個である。

第8図(a)の箱Tという名前のシンボル3は、突合せである。箱Tの伝搬元は例えば、箱Bという名前のシンボルである。箱Tの伝搬先は、例えば、箱Sという名前のシンボルである。

第9図(a)では、3行めのメモリに対応している。伝搬元、伝搬先変数には、突合せの場合、接続位置とシンボル名が格納される。

次に、第8図(a)の状態から第8図(b)のような状態へ変更することを考える。

「名前が箱Fで、種別がプログラムブロックで、プログラム名はMYFUNCで、入力変数はDIN1、DIN2の2個で、出力変数はOUT1の1個で引数変数はP1の1個であるようなシンボルを配置する」という意味の命令を、入力装置例えばタブレットを用いて入力する。

第3図の前処理ステップ210は、入力データから、モデル入力命令の種類がシンボル配置であること、名前が箱Fであること、種別がプログラムブロックであること、前述のようなプログラム

処理内容として、MYFUNCという名前のFORTRANプログラムをユーザが定義している。入力変数はDIN1、DIN2、出力変数OUT1、引数変数P1である。

以下において、第1図(a)のようなブロック線図を作成するために、第8図(a)から第8図(b)の、第8図(b)から第8図(c)へ変更していくという方針で説明する。

はじめに、第8図(a)の状態を考える。この時、第9図(a)に示すシンボルマップが作成されている。

例えば、第8図(a)のIN1という名前のシンボル1は、ランプ入力である。他のシンボルと接続されていない。第9図(a)では、1行めのメモリに対応している。伝搬元、伝搬先変数の個数は、まだ0個である。

第8図(a)のIN2という名前のシンボル2は、定数入力である。他のシンボルと接続されていない。第9図(a)では、2行めのメモリに対応している。伝搬元、伝搬先変数の個数は、まだ

名、入力個数、出力個数、引数個数などを生成する。

判断ステップ220で、シンボル配置に対応する処理へすすむ。

種別がプログラムブロックなので、ステップ230では新しいプログラムブロックのシンボル用のメモリをわりあてる。シンボルマップは第9図(a)の最下段にある箱Bの行の下に新たに追加される。

ステップ240で、名前変数に箱F、種別変数にプログラムが代入される。種別がプログラムブロックなので、種別変数にプログラム名、入力個数、出力個数、引数個数のメモリがわりあてられており、それぞれMYFUNC、2個、1個、1個の値が代入される。

ステップ240を実行した結果、シンボルマップに追加した新しいシンボルのメモリの内容が変更されて、第9図(b)のようになる。

後処理ステップ290を実行すると、表示画面が、第8図(a)から第8図(b)のように更新

される。

次に、第8図(b)から第8図(c)のような状態へ変更することを考える。

「IN1という名前のシンボル1と箱Fとを、結線する」という意味の命令を入力すると、第3図の前後処理ステップ210は、入力データから、モデル入力の種類がシンボル結線であること、入力端子がIN1であること、出力端子が箱Fの1番目の入力変数であること、などを生成する。

判断ステップ220で、シンボル結線に対応する処理へすすむ。ステップ250からステップ260を実行すると、シンボルマップ250は第9図(c)のように更新される。

ステップ250で、シンボルマップに含まれるメモリの名前変数が、IN1に一致するメモリを捜すためIN1が見つかるまで名前変数を順次サーチしていく。第9図(b)の名前の欄の1行目がIN1なので、その行の伝達先変数を1個新しくわりあてて、箱Fが代入される。

ステップ260で、シンボルマップに含まれる

目が箱Tなので、その行の伝達先変数を1個新しくわりあて、箱T上の入力端子の伝達元は箱Fの1番目の出力端子であるという意味の値が代入される。

以上のようなメモリの更新を行った結果、シンボルマップは第9図(c)のようになる。

後処理ステップ290を実行すると、表示画面は第8図(c)のようになる。

次に第1図(b)のようなイメージのプログラムブロックを作成するために箱Fの処理内容として定義されたプログラムを編集するという意味の命令を入力すると、プログラム編集ステップ270を実行し、この中でユーザが箱Fの処理内容として定義したプログラムを編集する。第8図(d)は表示の一例である。

以上の結果、第1図(a)のようなブロック図の作成が終わったので、モデル入力終了命令を入力し、前後処理ステップ210、判断ステップ220、終了ステップ280をとって第2図のステップ20の実行が終わる。

メモリの名前変数が、箱Fに一致するメモリを捜すため箱Fが見つかるまで名前変数を順次サーチしていく。第9図(b)の名前の欄の7行目が箱Fなので、その行の伝達先変数を1個新しくわりあてて、IN1が代入される。

同様に、「IN2という名前のシンボル2と箱Fの2番目の入力端子とを結線する」という意味の命令を入力すると、ステップ250で、第9図(b)の名前変数の2行目がIN2なので、その行の伝達先変数を1個新しくわりあて箱Fの2番目の入力端子が代入され、ステップ260で、第9図(b)の名前変数の7行目が箱Fなのでその行の伝達先変数を1個新しくわりあて、IN2が代入される。

同様に、「箱Fの1番目の出力端子と、箱Tの上の入力端子とを結線する」という意味の命令をすると、ステップ250で、第9図(b)の名前変数の7行目が箱Fなので、その行の伝達先変数を1個新しくわりあて、箱Tの上が代入され、ステップ260で、第9図(b)の名前変数の3行

次に第2図のステップ30が実行されると、第8図(d)で結線した箱Fというシンボル名のプログラムブロックの処理内容として、ユーザが定義したMYFUNCという名前のプログラムをコンパイルし、制御系CADシステムの全体プログラムとリンクする。

次にシミュレーションステップ40が実行される。完成したシンボルマップ第9図(c)が用いられる。

第5図の計算配列を作成するステップ410を実行すると、計算配列が第10図(a)(b)(c)の順に作成されていく。

第6図のステップ4110を実行すると、第9図(c)の種類の欄を順次調べていく。1行目がランプ入力なので、そのメモリの名前IN1を計算配列の名前の欄に登録する。同様にして、2行目が定数入力なので、IN2を計算配列に登録する。

ステップ4110を実行した結果、計算配列は第10図のように設定される。

次にステップ4120を実行する。シンボルマップ第9図(e)の名前の欄を順次調べていく。3行目の箱Tは計算配列第10(a)に登録されていないので、ステップ4130へすすむ。

次にステップ4120を実行する。シンボルマップ第9図(e)の名前の欄を順次調べていく。3行目の箱Tは計算配列第10図(a)に登録されていないので、ステップ4130にすすむ。

ステップ4130ではシンボルマップ第9図(e)の名前の欄について順次調べていく。はじめは1行目のIN1が試される。IN1は計算配列に登録済なので、判断ステップ4140によって何もしないでステップ4130へすすむ。次に2行目のIN2が試される。IN1と同様に、何もしないでステップ4130へすすむ。3行目の箱Tは計算配列に登録済なので、ステップ4150にすすむ。箱Tの伝搬元は箱Bと箱Fであるが、未登録なので、何もしないでステップ4130へすすむ。同様に4行目の箱Sは未登録だが伝搬元の箱Tが未登録なので何もしない。5

行目のOUTは、未登録だが伝搬元が未登録なので何もしない。

6行目の箱Bは未登録であるのでステップ4150へすすむ。全ての伝搬元箱Sは未登録だが、初期値は自力で決まる種類の積分であるためステップ4160が実行され計算配列に登録される。

7行目の箱Fは未登録で全ての伝搬元、IN1とIN2が登録済なので、ステップ4160が実行され、計算配列に登録される。

シンボルマップ第9図(e)について全部調べたので、判断ステップ4130を実行すると、ステップ4120へすすむ。

計算配列は第10(a)から第10図(b)のように更新されている。

再びステップ4120を実行する。シンボルマップ第9図(e)の名前の欄を順次調べていく。3行目の箱Tは計算配列第10(b)に登録されていないので、ステップ4130にすすむ。

前と同様に、ステップ4130でシンボルマッ

プ第9図(e)の名前の欄について順次調べていく。

IN1は登録済、IN2は登録済なので何もしない。

3行目の箱Tは未登録で、全ての伝搬元、箱Bと箱Fが計算配列第10図(b)に登録済なので、ステップ4160が実行され計算配列に登録される。

4行目の箱Sは未登録で、全ての伝搬元、箱Tが登録済なので、ステップ4160が実行され計算配列に登録される。

5行目のOUTは未登録で、全ての伝搬元、箱Bが登録済なので、ステップ4160が実行され計算配列に登録される。

6行目の箱B、7行目の箱Fは登録済なので何もしない。

シンボルマップ第9図(e)について全部調べたので、判断ステップ4130を実行すると、ステップ4120へすすむ。

計算配列は、第10図(b)から第10図(c)

のように更新されている。

三度ステップ4120を実行する。今やシンボルマップ第9図(e)の名前は全て登録済であるので、計算配列の作成が終了したことになる。完成した計算配列は、第10図(c)である。

第5図の計算配列を作成するステップ410の実行が終ったので、次にシミュレーション条件から、初期設定をするステップ420を実行する。

一例として以下のようなシミュレーション条件を入力した場合を考える。

計算開始時刻 0.5秒

計算終了時刻 1.5秒

計算きざみ 0.1秒

表示シンボル IN1, 箱F, OUT

これらの入力データを用いて、計算回数Nには11が、計算きざみDLTには0.1が、計算時刻TIMEには0.5が設定され、計算時刻と表示シンボルの各行をもった計算結果を示す図をわりあてる。

ステップ430を実行すると、計算結果を示す

図は第11図(a)のようになる。

初期計算時刻 0.5での初期値は、完成した計算配列第10図(c)を用いて求める。

計算配列第10図(c)のIN1、IN2は自力で今回値が定まる。今回値の欄に全て 0.0が設定される。箱Sの初期値を用いて、箱Bの初期値が定まり、今回値の欄に 0.0が設定される。IN1、IN2の今回値を用いて、箱Fの初期値が定まり今回値の欄に 0.0が設定される。箱B、箱Fの今回値を用いて、箱Tの初期値が定まり、今回値の欄に 0.0が設定される。箱Bの今回値を用いて、OUTの初期値が定まり、今回値の欄に 0.0が設定される。

こうして、計算配列の今回値の欄に、初期値が求められた。結果の必要なシンボルの初期値を、計算結果を示す図の1列目に複写後、次の計算の準備として計算配列の今回値を前四値の欄へ移動させる。

次に判断ステップ440を実行する。計算四数はまだ2回目なので、ステップ450へすすみ、

0.6秒における今回値は0.0となる。

判断ステップ4610を通過し、ステップ4620で、計算配列第10図(c)の3行目、箱Bを取り出す。ステップ4630が実行され、伝搬元箱Sの値0.0を計算配列第10図(c)から抽出し、ステップ4640を通過し、判断ステップ4650を通過して、分岐に対応した処理をすると、計算時刻0.6秒における今回値は0.0となる。

判断ステップ4610を通過し、ステップ4620で、計算配列第10図(c)の4行目、箱Fを取り出す。ステップ4630が実行され、伝搬元IN1の今回値0.1とIN2の今回値0.0を計算配列第10図(c)から抽出し、ステップ4640が実行され、自分自身の引数値をシンボルマップ第9図(c)から抽出し判断ステップ4650を通過して、プログラムに対応した処理ステップ4660を実行すると、プログラム名のHYFUNCを抽出し、ステップ4670でプログラムが実行され、計算時刻0.6秒における今回値

計算時刻を0.5から0.6に更新する。

次に計算ステップ460を実行する。

第7図のステップ4610で処理状況を判断しながら、計算配列第10図(c)について順次処理していく。

ステップ4620で計算配列第10図(c)の1行目、IN1を取り出す。自力で値が定まるのでステップ4630を通過し、ステップ4640の自分自身の引数値 1.0, 0.5, 1.5をシンボルマップ第9図(c)から抽出する。判断ステップ4650を通過して、ランプ入力に対応した処理をすると、計算時刻 0.6秒における今回値は 0.1 となる。

判断ステップ4610を通過し、ステップ4620で計算配列第10図(c)の2行目、IN2を取り出す。自力で値が定まるので、ステップ4630を通過し、ステップ4640の自分自身の引数値0.0をシンボルマップ第9図(c)から抽出する。判断ステップ4650を通過して、定数入力に対応した処理をすると、計算時刻

は0.5 となる。

判断ステップ4610を通過し、ステップ4620で、計算配列第10図(c)の5行目、箱Tを取り出す。ステップ4630が実行され、伝搬元箱Fの今回値0.5と箱Bの今回値0.0を計算配列第10図(c)から抽出し、ステップ4640が実行され、自分自身の引数値をシンボルマップ第9図(c)から抽出し、判断ステップ4650を通過して、突合せに対応した処理ステップ4690を実行すると、計算時刻0.6秒における今回値は0.5 となる。

判断ステップ4610を通過し、ステップ4620で、計算配列第10図(c)の6行目、箱Sを取り出す。ステップ4630が実行され、伝搬元箱Fの今回値0.5計算配列第10図(c)から抽出し、ステップ4640が実行され、自分自身の引数値と自分自身の前四値を抽出し、判断ステップ4650を通過して、積分に対応した処理を実行すると、計算時刻0.6秒における0.05となる。

ステップ460は前回値と今回値との差が無視できるとみなせるような計算きざみで計算する方法なので、箱Bを計算するのに用いた値との誤差は、無視できる。

判断ステップ4610を通過し、ステップ4620で、計算配列第10図(o)の7行目、OUTを取り出す、ステップ4630が実行され、伝達元箱Bの今回値0.0を計算配列第10図(o)から抽出し、ステップ4640を通過し、判断ステップ4650を通過して、と白出力に対応した処理ステップ4680を実行すると、計算時刻0.6秒における今回値は0.0となる。

以上により、計算配列の今回値に2回の計算結果が求められた。結果の必要なシンボルの値を計算結果を示す図の2列目に複写し、次の計算の準備として、計算配列の今回値を前回値の欄へ移動させる。

次に判断ステップ440を実行する。今度は3回目の計算になるのでステップ450へすすみ、計算時刻を0.6から0.7へ更新する。

も以下のような多くの方法が本発明の実施例から容易に推定できる。

プログラム名をあとから定義することができ、また、入力係数、出力係数をユーザがプログラムブロックに対して行った結線状況を用いて自動的に生成することもでき、引数係数をあとから定義することもできる。

更に、ユーザが引数係数の内容を計算時刻ごとに参照又は書き換えることができる。尚、書き換えにおいては、ユーザが作成したプログラム中において、引数係数に代入する値を定める代入文を挿入することによつてなされる。

(発明の効果)

本発明によれば、任意の制御動作を記述し、新たな制御要素として登録することにより複雑な動作を行なう制御対象を表現できるので、制御系を解析できる範囲が拡大される効果がある。

4. 図面の簡単な説明

第1図は本発明の一実施例の動作を説明する図、第2図は本発明の一実施例の全体の流れ図、第3

上記と同様に、計算ステップ460を実行する。

11回目の計算が終了すると、ステップ470により、計算結果を示す図第11図(c)のように、11列目計算終了時刻1.5秒における計算結果IN1.0、箱Fが1.0、OUTが0.5が格納される。判断ステップ440が実行され、第2図のシミュレーションステップ40の実行が終了する。

本発明によれば、以下に示すことが容易に推定することができる。

ユーザが任意の制御動作を記述する際においてFORTRAN 以外のプログラミング言語により記述することができる。また、プログラミング言語以外にも、例えば電気回路図やPAD図などより記述する場合においても本発明の実施例より容易に推定することができる。

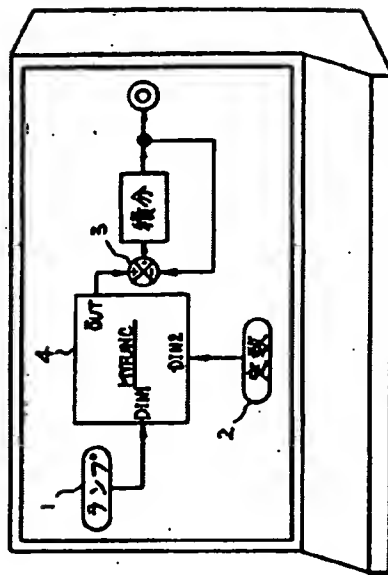
ユーザが任意動作を記述する際にプログラムブロックの処理内容として定義したプログラムの属性を保持する方法については、プログラムブロックを配置する際に、ユーザが定義する方法以外に

図はモデル入力ステップの詳細な流れ図、第4図はシンボルマップのデータ構造を説明する図、第5図はシミュレーションステップの詳細な流れ図、第6図は計算配列作成ステップの詳細な流れ図、第7図は設定された時刻における計算ステップの詳細な流れ図、第8図は本発明の一実施例を採用的に説明する図、第9図はシンボルマップを示す図、第10図は各シンボルの計算の順序及び前回値と今回値の配列を示す図(計算配列)、第11図は計算結果を示す図である。

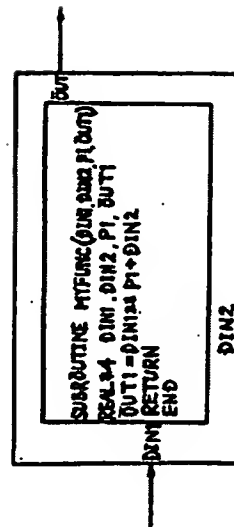
4…プログラムブロックのシンボル箱F、20…ブロック構造を作成するモデル入力ステップ、40…計算条件を決めて計算を行うシミュレーションステップ、270…ユーザがプログラムブロックの処理内容として定義したプログラムを編集するステップ、460…設定された計算時刻における計算を行なうステップ、4670…該当するプログラムを実行し、計算結果を格納するステップ。

代理人 弁護士 小川勝男

第 1 図
(a)

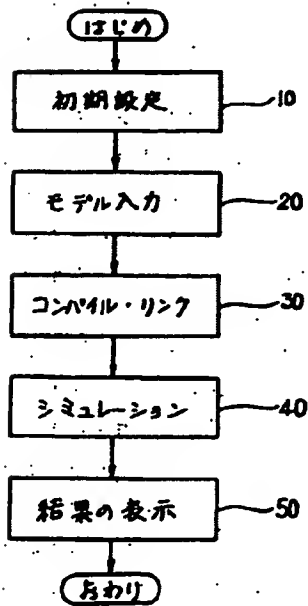


(b)



1... プログラム入力
2... 変数入力
3... 変数出力
4... プログラム出力

第 2 図



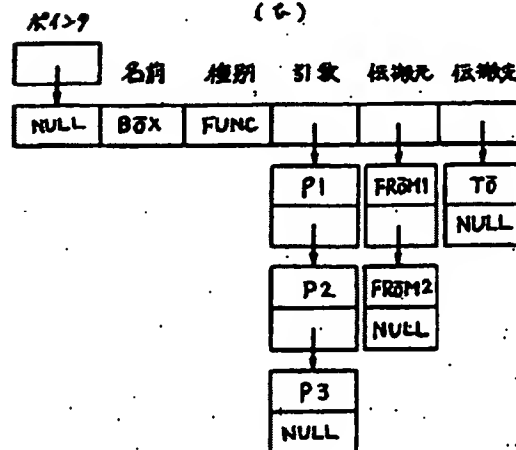
20... プログラムを作成するモデル入力ステップ
40... 計算条件を決定して計算を行うシミュレーションステップ

第 4 図

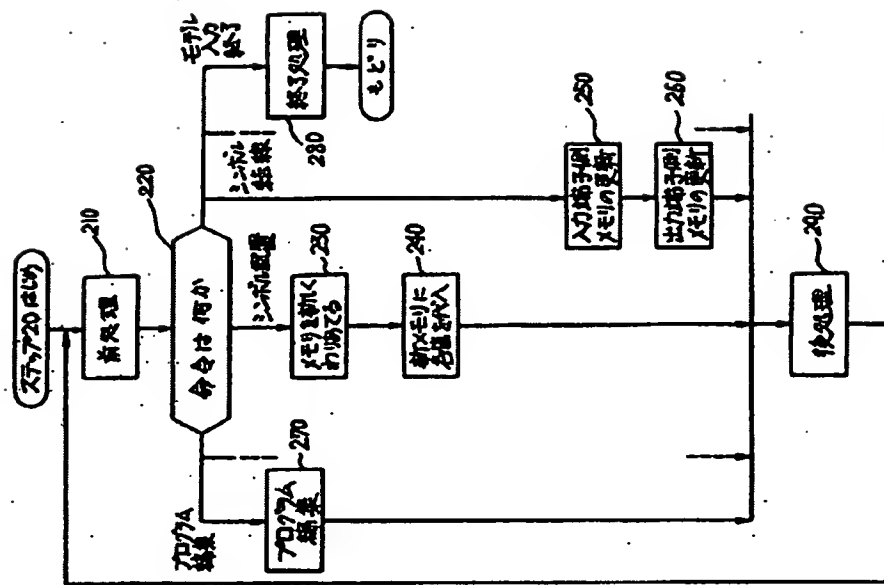
(a)

名前	種別	引数	伝搬元	伝搬先
B0X	FUNC	P1	FROM1	T0
		P2	FROM2	
		P3		

(b)



四三六



270...ユ-ザがアロギムボツク処理
内装して奥書(たアロギム)に
納まるるスダツア

第 4 圖
(c)

名前	種別	引数	拡張元	拡張先
	TOPA			

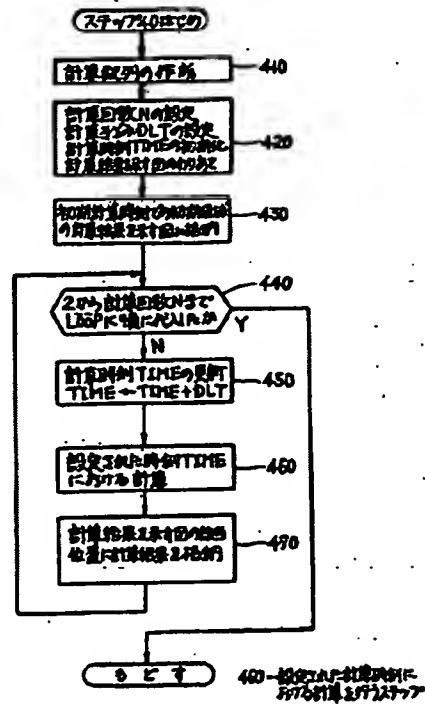
Page 2 (a)

```

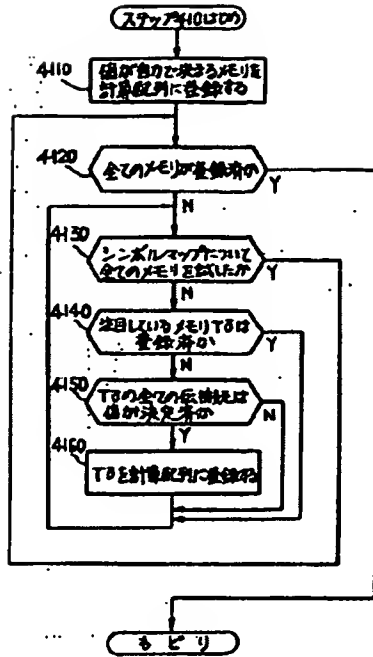
graph TD
    Root[ ] --> Name[名前]
    Root --> Gender[性別]
    Root --> Age[年齢]
    Name --> PRBC[PRBC]
    Gender --> 2[2]
    Age --> 1[1]
    PRBC --> 2[2]
    2 --> 1[1]
    1 --> 1[1]
    PRBC --> NULL1[NULL]
    2 --> NULL2[NULL]
    1 --> NULL3[NULL]
  
```

Figure 1 illustrates a hierarchical tree structure. The root node is 'NULL'. It branches into three nodes: '名前' (Name), '性別' (Gender), and '年齢' (Age). '名前' branches into 'PRBC'. '性別' branches into '2'. '年齢' branches into '1'. 'PRBC' branches into '2'. '2' branches into '1'. '1' branches into '1'. 'PRBC' also branches into 'NULL'. '2' also branches into 'NULL'. '1' also branches into 'NULL'.

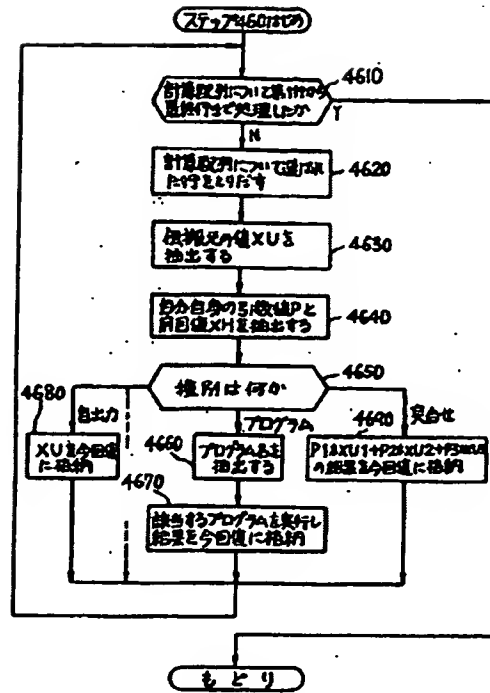
第 5 回



第 6 図

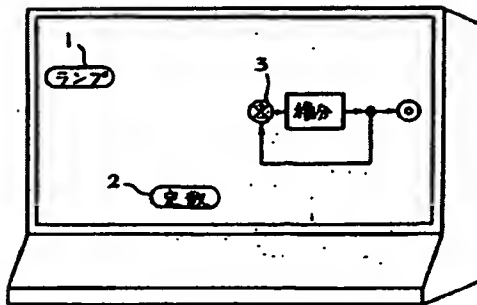


第 7 図

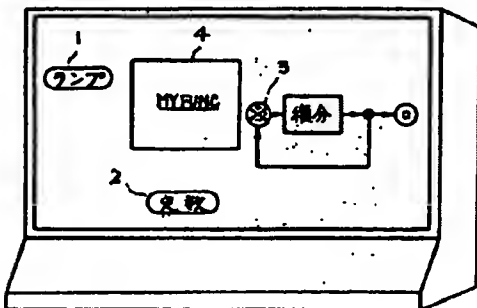


4670 ... 抽出するプログラムと一致するプログラムがあるか

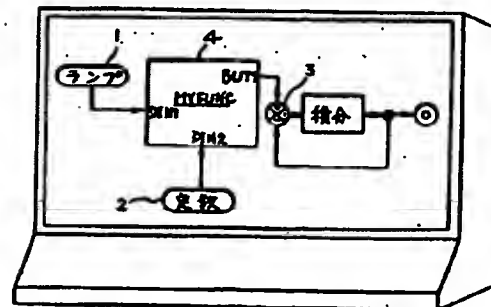
第 8 図
(A)



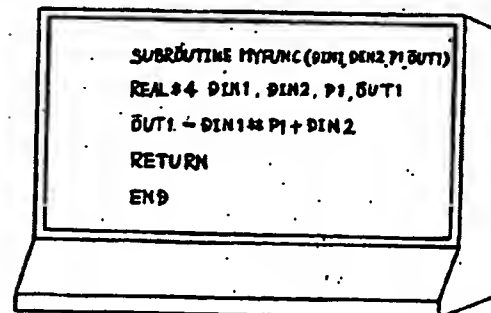
(B)



第 8 図
(C)



(d)



第 9 図

(a)

名前	種別	引数	伝搬元	伝搬先
IN1	ランパ	1.0	NULL	NULL
		0.5		
		1.5		
IN2	定数	0.0	NULL	NULL
箱T	乗合せ	0	下は箱B	右は箱S
		0		
		0		
		0		
箱S	積分	1.0	箱Tの右	箱B
OUT	出力	NULL	箱B	NULL
箱B	分岐	NULL	箱S	OUT
				箱TのF

第 9 図

(b)

名前	種別	引数	伝搬元	伝搬先
IN1	ランパ	1.0	NULL	NULL
		0.5		
		1.5		
IN2	定数	0.0	NULL	NULL
箱T	乗合せ	0	下は箱B	右は箱S
		0		
		0		
		0		
箱S	積分	1.0	箱Tの右	箱B
OUT	出力	NULL	箱B	NULL
箱B	分岐	NULL	箱S	OUT
				箱TのF
箱F	力/A	0.25	NULL	NULL
	MYFUNC			
		2		
		1		
		1		

第 9 図

(c)

名前	種別	引数	伝搬元	伝搬先
IN1	ランパ	1.0	NULL	箱Fの1
		0.5		
		1.5		
IN2	定数	0.0	NULL	箱Fの2
箱T	乗合せ	+1	下は箱B	右は箱S
		出口	上は箱F	
		-1		
		0		
箱S	積分	1.0	箱Tの右	箱B
OUT	出力	NULL	箱B	NULL
箱B	分岐	NULL	箱S	OUT
				箱TのF
箱F	力/A	0.25	IN1	箱Tの上
	MYFUNC		IN2	
		2		
		1		
		1		

第 10 図

(a)

名前	前日値	今日値
IN1		
IN2		

(b)

名前	前日値	今日値
IN1		
IN2		
箱B		
箱F		

(c)

名前	前日値	今日値
IN1		
IN2		
箱B		
箱F		
箱T		
箱S		
OUT		

第 11 図

(A)

名前	1回め			
計算時刻	0.5			
IN1	0.0			
給F	0.0			
OUT	0.0			

(B)

名前	1回め	2回め		
計算時刻	0.5	0.6		
IN1	0.0	0.1		
給F	0.0	0.5		
OUT	0.0	0.0		

(C)

名前	1回め	2回め		11回め
計算時刻	0.5	0.6		1.5
IN1	0.0	0.1		1.0
給F	0.0	0.5		1.0
OUT	0.0	0.0		0.5